100 mm
A5
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
1.0 mm
1.5 mm
2.0 mm
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

DOCUMENT RESUME

ED 301 186 IR 013 526

AUTHOR Masturzi, Elio R.
TITLE SEF.ASSISTANT: Shell for Authoring Language SEF.PC.
INSTITUTION Naples Univ. (Italy).
PUB DATE Mar 87
NOTE 36p.; Paper presented at the C.B.E. Methodologies Convention (Segrate, Italy, March 1987).
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Authoring Aids (Programing); *Computer Assisted Instruction; Computer Software; *Courseware; Foreign Countries; *Instructional Development; Microcomputers; *Programing Languages
IDENTIFIERS *Software Development

ABSTRACT

This paper describes SEF.ASSISTANT, a special software shell for authoring languages to be used in the development of courseware serving computer based education (CBE). The capabilities of SEF.ASSISTANT are identified: (1) it will generate programs written in the IBM SEF.PC authoring language; (2) it is compatible with external standard software (such as LOTUS and dBase III) as well as customized software; and (3) it can be used to drive videotapes and other multimedia. The functions performed by the software shell are described, and the use of the shell and the authoring language in coordinated procedures to construct courseware are detailed. The text is supplemented by three figures, one chart, and a SEF.ASSISTANT Quick Reference Sheet. (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED301186

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Elio R. Masturzi(*)

SEF ASSISTANT

SHELL FOR AUTHORING LANGUAGE SEF.PC

C.B.E METHODOLOGIES CONVENTION

IBM, Italy Spa, Segrate, March, 1987

(*)Department of Production Engineering,
University of Naples,Italy
Professor of Industrial Production Techniques,
Faculty of Engineering, University of Naples
Professor of Industrial Plant Management, Faculty
of Engineering, University of Rome

Department of Production Engineering,Piazzale Tecchio,
80125, Naples, Italy

IR013536

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Elio Masturzi
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

1) AIMS AND APPLICATION FIELDS IN C.B.E. TEACHING METHODOLOGIES

Computer Based Education (CBE) is a new teaching methodology destined to become rapidly diffused, as well as to profoundly change the existing traditional methods of training.

The need for training seems to grow daily.In order to evidence this, we need only note, that in the beginning of the next decade, there will probably, in the fields dealing with advanced technology (or advanced services) at least 300,000 persons. To this number, one must also add the number of current employees who must be newly trained in basis of the new tasks which they are assigned.

The need to develop new training courses for such a high number of people in such little time makes the use of new teaching methodologies, which use the new technology at hand indispensable.

CBE methods are part of the "Educational Technology" which uses in a coordinated and integrated manner a vast and articulated set of communication and teaching tools ; computers, videodisks, videotapes, projectors, televisions, etc.

As is already known, CBE is based on instruction principles which are developed using "programmed frames" which are administered to the learner according to a Learning Trail which varies according to the ability of each learner.

These "teaching frames", or "learning frames",theorized originally by Skinner, are used in the case of CBE utilizing "video-pages" belonging to software specifically planned and designed for teaching, called "coursewares".

The video-pages of the courseware can be divided into two types:

- lecture pages;

- question and answer pages.

The techniques adopted by CBE can be classified into four distinct categories:

a) Tutorial : These are intended to furnish informatic in basis of the planned instructing principles.

b) Testing : These are intended to evaluate the comprehension level reached by the student.

c) Problem Solving : These follow the teaching aims through the analysis and resolution of problems.

d) Simulation : This gives the learner practise through the use of simulated models.

TABLE 1

ADVANTAGES IN THE USE OF CBE METHODOLOGIES

- Increased level of learner interest
- Familiarization with the computer
- Training of personal in the actual place of work
- An increased quality level of the courses
- Facility of keeping employees up to date with new techniques
- Courses adaptability to each individual learner
- The pointing out and recording of learning capacities.

The advantages to be had using CBE are shown in Table 1.

The use of the methodologies in question seems to be advantageous when :

- Apprehension and capacity levels are very different.
- When the course must be expressed in different languages.
- The turn-over of students is very high.
- The students live far away from the learning center
- Courses must be taught in short time periods, and in different places at the same time.

Two events have contributed to the success of CBE methodologies in teaching, and in professional formation:

- the diffusion of personal computers
- the development of Authoring Languages and Systems

The availability of low cost hardware has brought us into a new era of "distributed micro-computing" which has allowed the machine to enter into the workplace.

The development of basic softwares,(Systems and Authoring Languages) specifically designed for teaching purposes has made the development of coursewares much easier, and thus made it available to instructors who do not have an elevated knowledge of computer science.

The sectors in which CBE teaching methodologies have been principally used in Italy up till now are banking, industry, and in the hospitals.
In these sectors, CBE coursewares inherent to three fundamental teaching fields have been developed and profitably used:

- professional education
- procedural training
- operative training

In the professional education field, CBE coursewares have been developed pertaining to the fields which most interest companies and credit institutions.
At the same time, CBE methodologies have been used to train employees in the use of new procedures in companies and in banks.

The operative training concerned with the functioning of machines, with the maintenance of plants, etc., has also been realized using CBE methodologies, integrated with the appropriate video support units (videodisks and videotapes).

2) CONSTRUCTING A CBE COURSE

The construction of an educational courseware takes place in four phases, each one requiring a percentage of the total time spent on developing the courseware as is shown:

1) Design (20% of total time);

2) Analysis and Development (30% of total time);

3) Programming (40% of total time);

4) Testing and Evaluation (10% of total time);

In order to develop a courseware, the combined effort of a team, made up of the following persons, is required :

1) An expert in the problem area;

2) An expert in teaching and in CBE methodologies;

3) A CBE analyst;

4) A CBE programmer.

To the first two experts will be given the duty of developing the courseware's "text", which will have to be done in a way completely different fom that which is used for a traditional text.

Past experience says that, in order for a CBE courseware to be done well, it is not possible to refer directly to a text already in existence; the "script" will have to be written taking into account the various didactical support systems which will then be utilised.

Having completed the writing of the text, the CBE analyst must choose the software that is to be used to develop the courseware's flow - chart, in function of the characteristic's of the Authoring Language which has already been chosen.

The programmer then has the task of translating the flow-chart into a set of instructions written in the Authoring Language which he wants to utilize.
Once that the courseware has been completed, it's validity as a teaching tool must be tested, by using it in one or more classes of students.

The realization of a courseware of approximately 200-250 video pages, equal to about 4 or 5 hours of teaching, will usually need about four or five hundred hours of analysis and programming ; therefore, anything which reduces the programming time is welcomed.

The use of a "shell" for Authoring Languages intended to eliminate, or at least to reduce the long and complex task of writing the instructions is very convenient.

SEF.ASSISTANT constitutes a Shell which has been specifically conceived in order to automatically generate blocks of instructions in the Authoring Language SEF.PC for IBM Personal Computers which work with the MS-DOS operating system.

3) AUTHORING LANGUAGES AND SYSTEMS

Authoring Languages and Authoring Systems are softwares characterized by a set of instructions which are conceived for the production of teaching programs.
Introduced in the beginning of the 1960's to be used with Main Frames, they have become diffused only with the introduction of Personal Computers. Since a great number of these softwares are available today, most of them work with the MS.DOS operating system.

Authoring Languages and Authoring Systems (A.L. and A.S.) usually have an "Editor" for text, and a graphic "Editor" through which it becomes possible to write and elaborate the text, and develop the eventual graphics needed.

Many languages and systems allow one to work with other programs written in other languages (BASIC,PASCAL,C, etc.). The more advanced ones also have a sub-set of instructions which are able to control a vast range of peripherical units such as videotapes, videodisks, projectors, etc.

The principal difference between an Authoring Language and an Authoring System is in the number of instructions, and in the facility of use.

Authoring Languages are programming languages, and in this sense, they are characterized by a set of syntaxical and grammatical rules which the programmer must learn and remember in order to use the language.

Since it is a language, the programmer usually has a large and varied set of instructions which then allow him to develop complex didactical programs.

However, the disadvantage in this is that one needs to have a solid background in both computer science and in programming.

The potential Author will not always have this background; and when this happens, he will not be able to personally develop the CBE courseware - he will have to then call in CBE analysts and programmers.

To avoid these inconveniences, Authoring Systems have been developed. These are much simpler to use than Authoring Languages, but their use is also much less flexible.

Authoring Systems have a limited number of microinstructions through which the author can develop the desired courseware. These microinstructions are usually not very numerous, and they are also usually easy enough to remember and to understand.

Obviously though, when you must develop a complex courseware, you may encounter problems using a set of instructions which is rigid and limiting.

In light of these considerations, we can see why a Shell for an Authoring Language has been developed.
This Shell allows you to combine the creative characteristics of an Authoring Language and the simplicity of use of an Authoring System.

Among the languages and the systems available on the market, we will concentrate on those which can be used with Personal Computers and the MS.DOS operating system.

The softwares which we used, and which we feel should be mentioned, are the following:

1) MIAS Mc Graw Hill;

2) PC-PILOT Washington Computer Service;

3) SEF-PC IBM.

The first is an Authoring System which is simple enough to use, and which has an excellent text editor. It's graphic utilities are also very well developed and of good quality. On the other hand, the use of MIAS is very laborious once the Author has become familiar with the software.
MIAS contains a set of instructions which can control the videotape, but cannot be linked to external programs by the Author.

The PC-PILOT, the MS-DOS version of PILOT, is an Authoring Language which has a set of very powerful instructions which are though very difficult to understand. This can then be used with external programs written in other languages, and it can also be used with other languages and control peripherical units such as videotapes and, in it's latest version, also video-disks.

SEF.PC is IBM's Authoring Language.
It's use is characterized by the specific advantages of the language, but also by it's difficulty, which is considerably higher than that of Authoring Systems.
We will now go on to examine and illustrate the fundamental characteristics of the SEF.PC Authoring Language, and then the "tool kit" SEF.ASSISTANT, specifically designed to work with it, will be spoken about.

4) THE SEF.PC AUTHORING LANGUAGE

We will now examine the operative logic and structure of the SEF language in order to better examine the aims which we had in mind when developing the Shell SEF.ASSISTANT.
In any case, the manual must be examined if one wants a detailed analysis of the software in question.

SEF.PC is an Authoring Language specifically developed to work with IBM Personal Computers and the MS-DOS operating system.

In it's most recent version, the SEF-PC is composed of eight programs which can essentially be divided into the following five categories:

1) An Editor program, (XTED) which allows the Author to write, and/or change the individual video pages , and also to develop a "learning tree" of the expected learning trail.

2) A Test program, (SEFAUT) which allows the author to test the program which he is developing.

3) A cryptographic program (ENCODE) of the program developed.

4) SEFSTU, a program which allows the student use the courseware developed.

5) Programs containing utility functions and "help" messages for both the Author and the Learner.

SEF requires that the instructions relating to each video page be recorded in the same sequence as the individual pages are called up.

The set of instructions are recorded in a file

AAABBBOO.XXX

in which the letters:

XXX: represent the name given to the course

AAA: represent the name given to a single chapter

BBB: represent the name given to a single paragraph in a chapter.

SEF does not have an index for the single video pages.

A SEF courseware allows one to "address" the single paragraphs of the course using a menu with two levels, representing, respectively the general index and the chapter index.
In Fig.1, we can see the general structure of an SEF course.

4.1) SEF.PC VIDEO PAGES

SEF.PC runs through a "teaching trail" through the visualization of a series of three types of video pages:

1) Menu Pages;

2) Lecture Pages;

3) Question and Answer Pages.

The Menu video pages obviously serve to indicate the individual arguments dealt with in the courseware.
The lecture pages contain the texts with the information which one intends to furnish.
The question and answer pages contain the questions which the Author wishes to ask along with the introduction fields of the answers.

On both the lecture pages and the question pages it is possible to have "windows" opened which can contain messages or suggestions.

The question pages may be distinguished, in function of the number of fields foreseen for the answer, as:

1) a question with only one field of input;

2) a question with multiple fields of input.

The question pages may contain:

- suggestion messages (maximum 9);

- the exact answers;

- messages (maximum 9) corresponding to the various types of erroneous answers foreseen by the author.

According to which type of page has been chosen, the messages and suggestions appear in different zones of the video screen.

The windows, which may be positioned wherever on the screen, open after a pre-fixed "waiting time" and close after a pre-fixed "viewing time", both of which are pre-established by the author.

SEF's video pages can be formatted differently (standard, free, graphic, etc.), and the number of characters per line may be equal to forty or eighty characters (graphic or textual).

SEF allows the Author to develop the lecture pages as well as the question and answer pages using various colors for the text itself as well as for it's graphic accompaniments.
The coordination of the background and of the text's colors can be established time after time by the author in basis of the palette of standard colour couplings made available by SEF.

The background colors and the character colors depend upon the type (explaining, action, error, result) of window which one wishes to open.

The text of the various pages is elaborated using the X.TED Editor with which are also written the instructions necessary to "shape" the single pages or the single windows.

In Fig.2, we can see the standard block of SEF instructions which are concerned with the configuration and the text of a lecture page.

The block of instructions relating to a particular window must be immediately inserted after the block of instructions relative to the video pages upon which one wishes to open the window.

In Fig.3, we can see the standard block of instructions which are necessary in order to construct a window; in this sense, it is necessary to establish the text of the window as well as all the instructions (waiting time, visualization time, closing time, etc.) inherent to it's use.

The question and answer pages must be developed according to the following format.

- displaying of the text;
- definition of the number and sequence of the suggestions;
- definition of the exact response;
- definition of the "Editor" form (elimination of blank characters, search for a particular string of characters, etc.) to apply during the reading phase, as well as the scanning of the string of characters corresponding to the response given;
- definition of the type of controls to extend to the answers;

- definition of the eventual wrong answers foreseen;

- definition of the messages corresponding to the erroneous answers foreseen.

In Fig.4, we can see an example of a sequenced block of SEF instructions which are necessary in order to construct a question page of the "single input field" type.

As we have already said, the question page may be configured in a way so as to have more input fields for the answers. Up to ten fields can be had for each page. In this case, the position of each single field must be established by the Author when he is constructing the page.

4.2) SEF.PC COMMANDS AND EDITOR

The SEF Authoring Language has a set of Editor commands which is capable of:

- assigning a name to a file;
- saving, erasing, and loading a file;
- printing;
- adding on, erasing,and copying lines or blocks of lines;
- inserting one file into another.

Among other things, SEF's set of instructions allows one to:

- To append the current file to one already saved (APPEND).

- To jump from one line of instructions to another (BRANCH) which has been identified using a LABEL. This is similar to BASIC's GO TO command.

- To jump from one line of instruction to a block of instructions (which will be eventually contained within a another file) , execute these instructions, and then go back to the instruction immediately following that one from which you started. This is similar to BASIC's GO SUB cqmmand.

SEF also allows you to develop graphics, as well as letting you link them to external files (data or programs).
So then, it is possible to use SEF with external programs written by the author in different languages (BASIC,PASCAL,etc.) which are then given tasks such as developing calculations, creating or manipulating images, etc.

Lastly, SEF allows you to control external peripherical units, such as video-disks so that you can use them as teaching aids when the courseware is being used.

5) THE SEF.ASSISTANT SHELL

Even though we have spoken only briefly about the Authoring Language SEF,it is still possible to formulate the following outline.

- SEF is an Authoring Language, and because of this, it's use requires that one apprehend and memorize a fairly high number of syntaxical and grammatical rules, as well as commands, instructions, and microinstructions.

- The construction of the video pages is rather tedious, and requires a lay out of the screen, through which one tells the system all the information, and the "attributes" characteristic of the singular page.

- While one is writing the block of instructions relative to a single page (or window), it is not possible to immediately check what one is doing . So then, eventual errors or variations in the text, in it's position, colors, etc., will come to light, and so then be corrected, only when the courseware is tested.

- It is not possible to give a single video page, or a single window an identifying code. To modify or erase something from a single page, one must look at the complete listing of the file AAA BBB OO. XXX which contains the page in question, modify the block of instructions inherent to that page, and so then repeat the test run.

SEF.ASSISTANT software has been specifically developed to avoid certain inconveniences present in SEF-PC, and to make possible the use of this Authoring Language to users who are not experts in the fields of Computer Science and programming.

The SEF.ASSISTANT can be called a "shell" able to automatically generate a courseware in SEF.
So then, the "shell" enables the author to construct the single video pages in an extremely simple and rapid manner, without having to use SEF's grammatical and syntaxical rules.

5.1) SEF.ASSISTANT EDITOR

The construction of pages or of windows, the writing of the texts, the design of graphics, etc. are all done through the EDITOR program.

It extensively uses functional keys, and this makes it extremely easy to use.

With the EDITOR program, it is possible to:

- create and vary characters and graphics;
- construct graphic figures such as frames, boxes, lines, etc.;
- manage color palettes, etc.

The video pages which have been created, and whose images are always present upon the screen, are automatically translated into SEF when the Author has established what is to be the lay out and the contents of the screen.

SEF.ASSISTANT writes by itself all the SEF instructions necessary to develop the page desired by the Author, and so then creates less programming work.

5.2) FUNCTIONS PERFORMED BY SEF.ASSISTANT

SEF.ASSISTANT performs a great number of very useful tasks during the development stage of a courseware in the Authoring Language SEF.
Among the possible functions, we have chosen the following to emphasize:

A) CONSTRUCTION OF THE VIDEO PAGES

The Shell allows the construction of any type of page allowed by SEF :

- lecture video pages;
- question video pages;
- menu video pages.

These contain characters as well as graphics.

B) WINDOW CONSTRUCTION

The Shell has a set of macroinsstructions which are able to construct windows rapidly and easily, windows which will then be inserted into video pages which have already been written.

C) CONTINUAL CHECKING

This is a particularly important advantage of SEF, the fact that the video pages which the Author is creating are always present on the screen. This makes it possible to immediately review their contents as well as the positioning on the screen without having to then use the SEFAUT program.

D) CREATION OF VIDEO PAGES WHICH CAN BE EDITED

The pages created by SEF.ASSISTANT can be saved, and then recalled in an "editable" form. This makes it possible to erase, modify, etc., with the help of SEF.ASSISTANT's EDITOR before translating something definitively into SEF instructions.

E) SEF.ASSISTANT'S PAGE DATA BASE

The program Shell allows for pages created but not yet translated to be saved in a "SEF.ASS Screens Base". These screens can be picked up from the storage file question at any moment, modified using the EDITOR of the Shell and then they can be translated into their final version.

F) INSERTION OF THE ATTRIBUTES

For the construction of windows and of question pages, SEF requires that a set of instructions be written, which contain all the information necessary in order to create the window, or the question page.These instructions,which define both the window and the question pages, are called "attributes" by SEF.
SEF.ASSISTANT , requires only the guided input of the data in question, in fact, the Shell develops by itself the instructions which are necessary in the SEF language.

G) SEF PAGE DATA BASE

The video pages created and then translated with SEF.ASSISTANT can be individually saved in a "SEF Screen Base". With SEF,it is not possible to recall the individual pages which have previously been created. This can only be done by using the program XTED.

The Shell in question then foresees the use of two video page Data Bases.

1) SEF.ASS Screens Base : contains video pages which have still not been translated into SEF and are therefore modifiable by the Shell's EDITOR.

2) SEF Screens Base : This contains video pages translated into SEF, singularly identifiable with SEF's EDITOR.

H) CONSTRUCTION OF A LEARNING TRAIL

The video pages saved in one of the two Screens Base may be singularly picked up and linked together according to the Learning Trail which has been decided upon by the Author.

So then, it is possible to:

- Define the learning trail after the creation of the video pages;
- Construct different learning trails using the same sets or sub sets of pages.
- Construct diffent coursewares through the assembly of video pages which have precedently been saved in a "library" of graphic images and video pages.

5.3) COORDINATED USING PROCEDURE

In light of what has already been said about constructing a courseware through the use of SEF and of the SEF.ASSISTANT Shell, it will be useful here to demonstrate the logical coordinated procedure which the coursewares construction requires.(Fig.5)

This procedure consists of six work phases, the first five of which are done ina SEF.ASSISTANT environment and the last of which is done in a SEF. environment.

PHASE ONE)

The author constructs the needed video pages using SEF.ASSISTANT's EDITOR.

During this plan,the text he is working on is always visible. To develop these pages, the Shell EDITOR is used, and, as has already been said, it has a wide range of functions and utilities which are easily and quickly understood.

PHASE TWO)

The pages which are created are saved in the file "SEF.ASS Screen Base" so that they are individually identifiable and able to be recalled in RAM memory.

PHASE THREE)

Each page retained in the SEF.ASS Data Base can be reloaded into the central memory, checked ,and eventually modified using the utility functions of SEF.ASSISTANT's EDITOR.

PHASE FOUR)

The lecture pages, question pages and windows developed in the definitive form wished for by the Author are completed with the input of their attributes and of other information requested by SEF.
SEF.ASSISTANT then translates the single pages into blocks of SEF instructions.

The blocks of instructions concerning a single page may be individually saved in the "SEF Screen Assistant File".
The architecture of the SEF.ASSISTANT allows every video page contained in the Data Base to be singularly collected and modified, using the XTED program.
This cannot be done with SEF, which does not allow for the identification, and thus the selection of a single page.

PHASE FIVE)

In this phase, which does not necessarily coincide with that of the creation of the video pages, the Author establishes the learning trail.
The single pages translated into SEF are picked up from the SEF Screen library,linked and then inserted into the sequential file AAA BBB OO.XXX according to the SEF rules.
In this phase, it is possible to use both old and new video pages.
At the end of this phase, you have a courseware written in the SEF Authoring Language.

PHASE SIX)

The courseware obtained at the end of these five phases is now ready to be used in the SEFAUT program.

Eventual minor modifications may be done by correcting individual lines of instructions with the XTED SEF editor.

The definitive version of the courseware is then coded, using the ENCODE program. This is available in a "student disk" which the student should use along with the SEF's program SEFSTUD, when he is using the courseware.

6) CONCLUSION

The SEF.ASSISTANT Shell is a software specifically designed in order to be used alongside the Authoring Language SEF-PC.
It can be used on Personal Computers which use an MS-DOS operating system, with at least 256 KRAM and a color monitor.

The use of SEF.ASSISTANT allows the Author to develop coursewares without having to concern himself with programming. In fact, the Author is able to construct the single elements of a course in a very simple and intuitive manner which is then translated by itself into instructions in the SEF language.

The development of SEF.ASSISTANT was brought about through my personal experiences in both research and CBE teaching methodologies.
In my professional as well as teaching work, I have had to develop very extended coursewares, (each one contains over 1000 video pages) with very complex screens and which have to be linked with external programs written in BASIC. In this case, it was difficult to use Authoring Systems, and I had to use Authoring Languages.

Among these, I preferred SEF-PC because of it's technical characteristics as well as it's compatibility with other basic softwares (Story-Board, video-disk controller, etc.) and applicative softwares available for IBM Personal Computers.

SEF, however, is a language, and so then the development of extensive and complex programs requires the work of highly qualified personnel for long periods of time.

Because of this, I decided to plan and develop, along with TECHNOVA's technicians, a software which would simplify the use of SEF , making it as easy to use as an Authoring System.

The use of SEF together with the SEF.ASSISTANT Shell has enabled me to make coursewares in about one third of the time that would have been used had I been working with SEF alone.
The time used was also inferior to that needed with an Authoring System.

Another advantage in using SEF.ASSISTANT was that of having to use very qualified personnel only in the analysis, formulation and control phases; in the other phases, such as the in-putting of the course, it was possible to use less qualified personnel, and thus keep the production price of the course down.

The SEF.ASSISTANT software is distributed by Technova, Via Posillipo 130, 80122 Naples, (Italy).

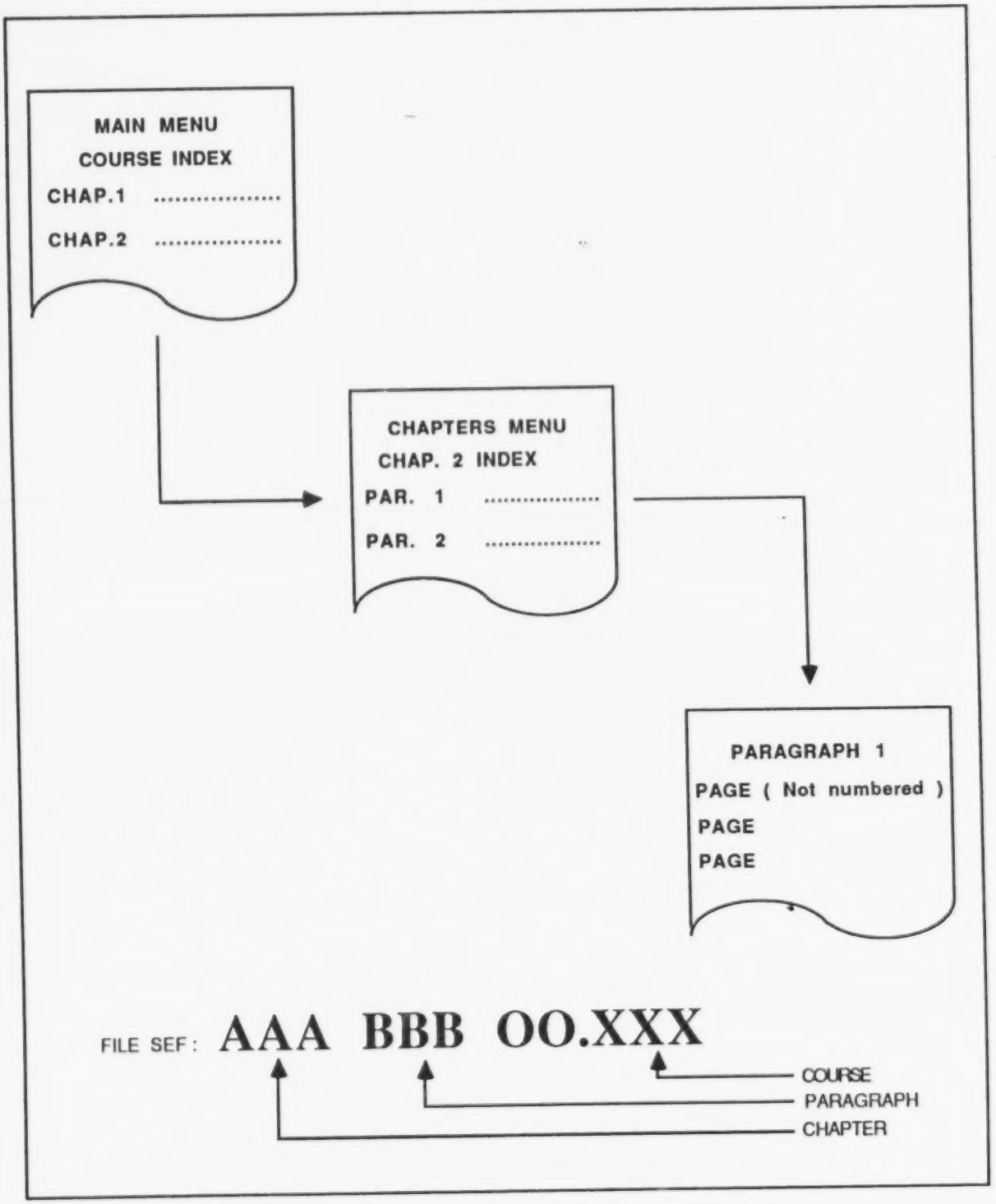

MAIN MENU
COURSE INDEX
CHAP.1 .....
CHAP.2 .....
CHAPTERS MENU
CHAP. 2 INDEX
PAR. 1 .....
PAR. 2 .....
PARAGRAPH 1
PAGE (Not numbered)
PAGE
PAGE
FILE SEF: AAA BBB OO.XXX
COURSE
PARAGRAPH
CHAPTER

FIG. 1

EXAMPLE : INSTRUCTION BLOCK LECTURE PAGE

! XYZ

New page command and statement
- page type
- back-ground color
- palette, etc.

.....

.....

.....

.....

Page text

.....

.....

Comments for the instruction block

A/I/SL/SW/SP/W

SEF's Commands (APPEND, IMBED, LOCATE, WAIT, PRINT, WINDOW, ECC.)

FIG. 2

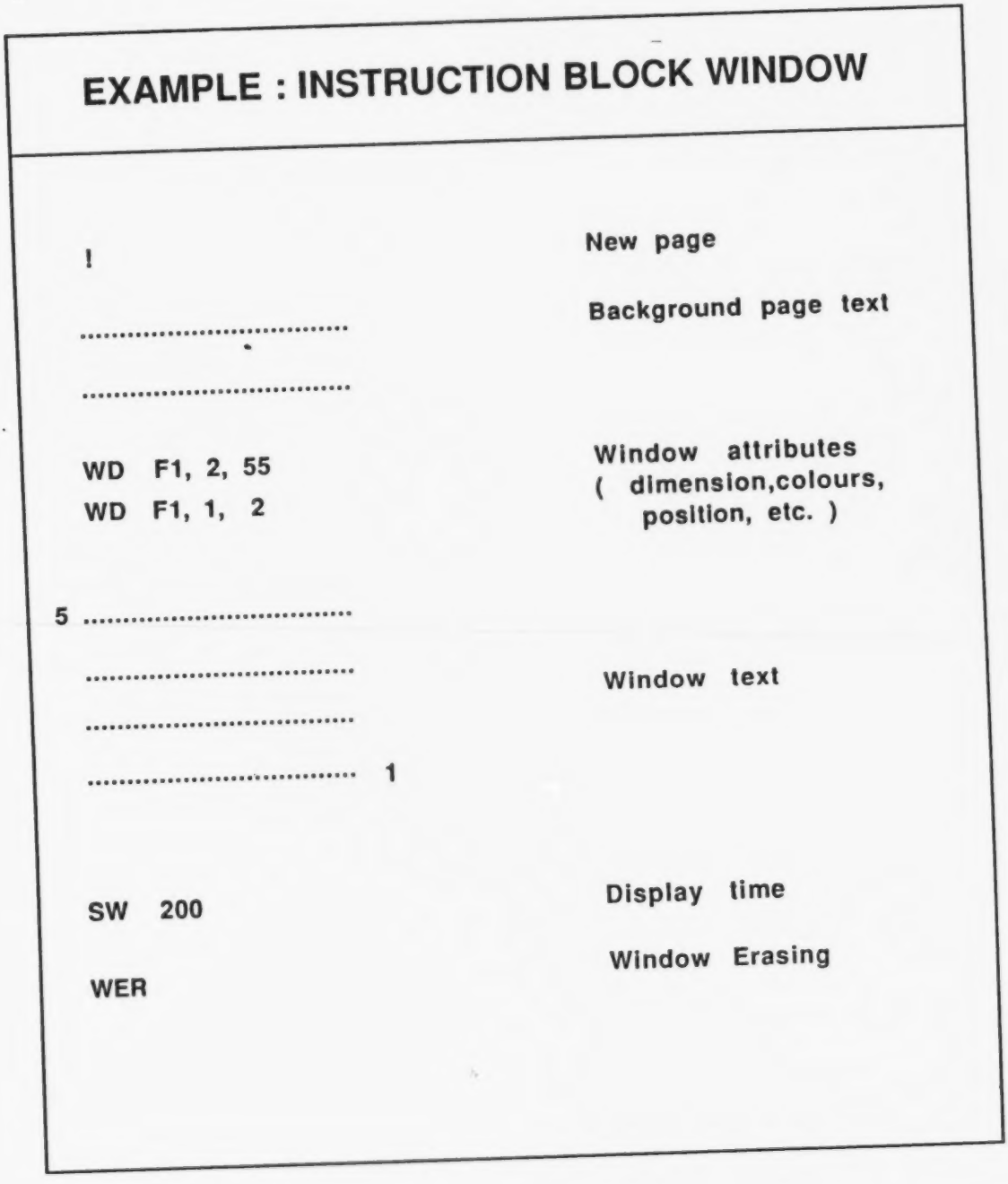

EXAMPLE : INSTRUCTION BLOCK WINDOW
!
New page
.....
Background page text
.....
WD F1, 2, 55
WD F1, 1, 2
Window attributes
(dimension,colours,
position, etc.)
5 .....
.....
Window text
.....
..... 1
SW 200
Display time
WER
Window Erasing

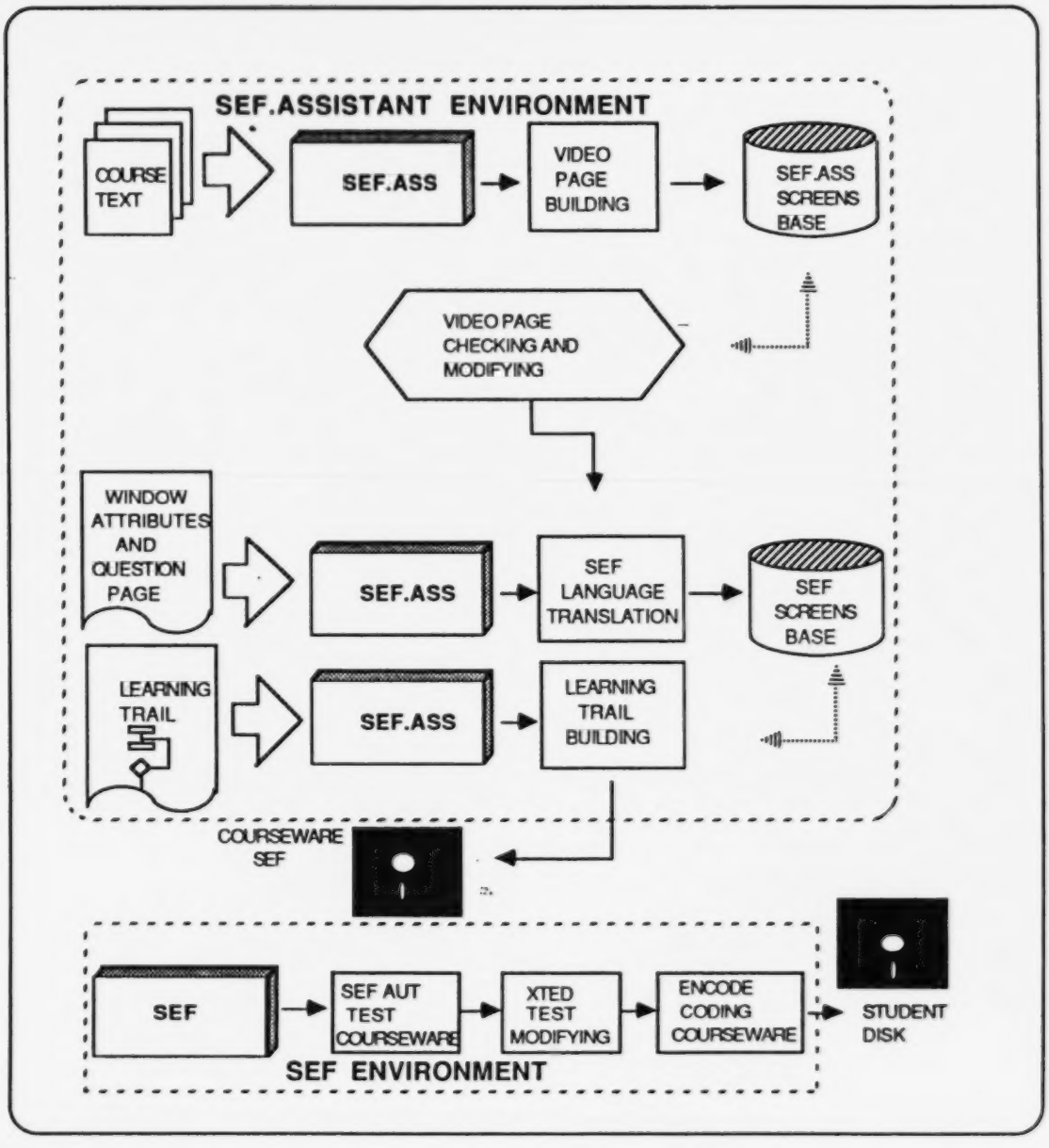
FIG. 3

EXAMPLE: INSTRUCTION BLOCK QUESTION PAGE

!	New Page
? X, Y / Z	Question Type
....	Question Text
H	Suggestions
=	Correct answers
E	Editor form for answers
+ (.....)	Wrong answer message
/ ..... - (.....) / ..... - (.....) /	Correct answer message Wrong answer 1 Wrong answer message 1 Wrong answer 2 Wrong answer message 2

FIG. 4

OPERATIVE STRUCTURE AND LOGIC OF THE SEF PC AUTHORING LANGUAGE



SEF.ASSISTANT® Shell for Authoring Language SEF-PC — Quick Reference Sheet

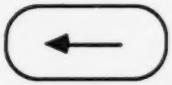
SPECIAL KEYS

To move the coursor

DEL — To cancel characters

INS — To insert characters

Coursor to the beginning


Left forward moving without erasing

HOME — Coursor to right upper corner

END — Coursor to left bottom corner

PgUp — (Pg.Graphics) Symbol Display

PgDn — (Pg.Graphics) Symbol Display

Quick moving (5 characters)

FUNCTION KEYS

F2 — Text/Graphic Page Commuting

F3 — Selection of colour couplings

F4 — Frames drafting

F5 — Working area erasing

F6 — Input fields positioning

F7 — Area painting

F8 — Page reading in EDIT form

F9 — Recording page

F10 — Translating

↑+F1 — Window building start up procedure

↑+F2 — Question character inserting

FUNCTION KEYS

↑+F3 — Horizontal Line drafting

↑+F4 — F4 Disactivation

↑+F5 — F5 Disactivation

↑+F6 — F6 Disactivation

↑+F7 — F7 Disactivation

↑+F8 — Activation Function APPEND

↑+F9 — Activation Function IMBED

ALT+F1 — Brushing

ALT+F2 — File name changing

ALT+F3 — Suffix .XXX changing

ALT+F4 — Data disk definition

ALT+Q — Return to Main Menu

ALT+C — Video page erasing in RAM

TECHNOVA S.r.l. - via Nuova S.Rocco, 62 - 80131 NAPOLI - Tel. 081/7419041

ERIC Clearinghouse on
Information Resources
Syracuse University
School of Education
Syracuse, N Y 13210
(315) 423-3640

AUTHOR ABSTRACT
(To Accompany Document Submitted to ERIC/IR)

Author: Elio Masturzi

Title: SEF.ASSISTANT shell for authoring language, SEF.PC

Abstract (up to 200 words):

It is illustrated a special software tool-kit , designed and achieved to make the author of a text able to write a courseware by himself.
The SEF.ASSISTANT is a powerful generator of programmes written in IBM SEF.PC authoring language.
It is compatible and linkable to external softwares standard (e.g. LOTUS, dBIII, etc.) as well as customized; the software allows to drive videotapes or other multimedia too.

Information About Docume...

Paper presented at:

Project Report (Contract Infor...ation, if possible)

Journal Article (Complete Cit... ion)--(Some journals do not appear in CIJE)

Other

IR 013586

BEST COPY AVAILABLE